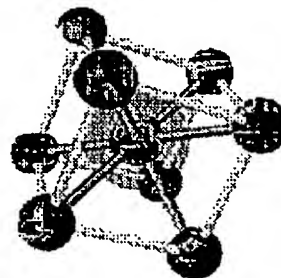


**AS-I Wizard Test Plan**

---

**Document C**

**SIEMENS**



**STEP 7-Micro/WIN 32**

***AS-I Wizard***

**Test Plan**

**BEST AVAILABLE COPY**

**AS-I Wizard Test Plan**

1 Introduction.....	3
2 Wizard Dialog Appearance and text.....	3
2.1 SCREEN 1.....	3
2.2 SCREEN 1A - PROGRAM COMPILATION.....	4
2.3 SCREEN 1B - INVALID CONTROLLER SCREEN.....	5
2.4 SCREEN 2.....	6
2.5 SCREEN 3.....	7
2.6 SCREEN 4.....	9
2.7 SCREEN 5 - DIGITAL SLAVE.....	10
2.8 SCREEN 6 ANALOG SLAVE.....	18
2.9 SCREEN 7.....	19
3 Reload tests.....	20
3.1 CONFIGURATION VERIFICATION.....	20
3.2 SCREEN 8.....	21
3.3 DELETING CONFIGURATION.....	22
3.4 MOVING CONFIGURATION.....	23
4 Compare tests.....	23
4.1 COMMAND AND RESPONSE BYTES.....	23
4.2 SLAVE MATCH/MISMATCH.....	23
4.3 COMMUNICATION DISRUPTION.....	23
5 Update tests.....	24
6 Code Generation tests.....	24
6.1 SYMBOL TABLES.....	24
6.2 POU ELEMENTS.....	24
6.3 POU COMMENTS.....	24
6.4 DATA BLOCK.....	24
6.5 IBC.....	24
6.6 ANALOG INPUT FILTERING.....	24
7 Functional tests.....	24
7.1 DISCRETE TRANSFER.....	24
7.2 EXTENDED ADDRESS B TRANSFER.....	25
7.3 ANALOG TRANSFER.....	25
7.4 AS-I READ.....	25
7.5 AS-I WRITE.....	25
8 Change slave address tests.....	25
8.1 APPEARANCE.....	25
8.2 NAVIGATION.....	26
8.3 PREVIOUS BUTTON.....	26
8.4 NEXT BUTTON.....	26
8.5 CANCEL.....	26
8.6 DELETE CONFIGURATION.....	26
8.7 MOVE CONFIGURATION.....	26
8.8 COMMUNICATIONS TESTS.....	26
8.9 SLAVE DETECTION.....	26
8.10 SLAVE MOVING.....	26
9 Ozzy.....	27
9.1 INTRODUCTION.....	27
9.2 HARDWARE REQUIREMENTS.....	27
9.3 SEQUENCE OF EVENTS.....	29
9.4 INSTRUCTIONS THAT ARE SUPPORTED BY OZZY INCLUDE.....	29

## AS-I Wizard Test Plan

### 1 Introduction

The AS-I Wizard is a new addition to Micro/WIN for release 3.2.1. This new wizard will simplify the process of using AS-I slaves within Micro/WIN.

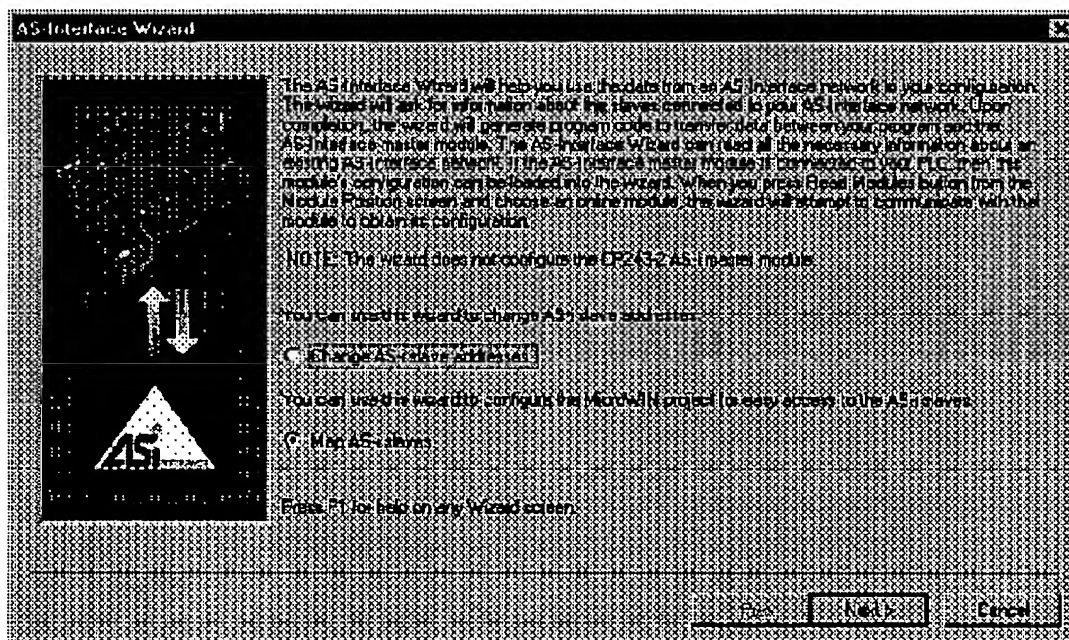
Not only does the wizard allow you to define symbolic names within your program, it also provides subroutines for data transfer and manipulation.

Also provided is a compare and update diagnostic tool to help in debugging your AS-I network. This plan outlines all areas within the AS-I wizard that must be verified before product release.

### 2 Wizard Dialog Appearance and text

#### 2.1 Screen 1

Like the modem wizard, the ASI wizard is two headed. From the initial screen shown below, you can either change a slave address or Map ASI slaves for use within your program.



#### 2.1.1 Appearance

Verify that the picture is correct and is located correctly within the window.

Verify that text is technically and grammatically correct.

Verify that no spelling errors exist.

Verify that no truncation problems exist,

#### 2.1.2 Navigation

Verify tab and arrow navigation is correct.

Verify that upon entry, the NEXT button is active

## AS-I Wizard Test Plan

---

### 2.1.3 Next button

Verify that the next button takes you to the correct next window

### 2.1.4 Cancel button

Verify that the cancel button brings up the cancel failsafe window.

Verify that exiting the failsafe with NO leaves you in the window.

Verify that exiting the failsafe with Yes correctly closes down the wizard.

### 2.1.5 Window close

Verify that the window close button has the same effect as the Cancel button

### 2.1.6 Change ASI slave addresses

When this radio button is active, the MAP ASI button should be inactive, these two buttons are mutually exclusive.

Verify that when this button is active, that the NEXT button takes you to the change slave address screen.

### 2.1.7 Map ASI slave

When this radio button is active, the Change ASI slave address should be inactive, these two buttons are mutually exclusive.

Verify that when this button is active, that the NEXT button takes you forward into the map slave window.

## 2.2 Screen 1A - Program compilation



### 2.2.1 Appearance

Verify that the picture is correct and is located correctly within the window.

Verify that text is technically and grammatically correct.

Verify that no spelling errors exist.

Verify that no truncation problems exist.

### 2.2.2 Navigation

Verify tab and arrow navigation is correct.

Verify that upon entry, the YES button is active

### 2.2.3 NO button

Verify that the NO button cancels the wizard

### 2.2.4 YES button

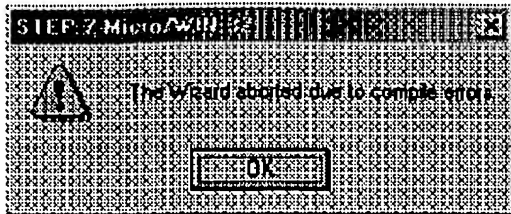
Verify that the YES button begins a program compilation

Verify that Passed compile takes you to the next screen (3)

Verify that failed compile brings up the following screen:

## AS-I Wizard Test Plan

---



Verify that Pressing OK, or hitting the X dismisses this window and closes down the wizard.

### 2.3 Screen 1B - Invalid controller screen

Verify that the Wizard generates the following screen whenever you select a CPU type that the wizard does not support. Verify that you are allowed to proceed to screen 1 when a supported type is selected.



#### 2.3.1 Appearance

Verify that the picture is correct and is located correctly within the window.  
Verify that text is technically and grammatically correct.  
Verify that no spelling errors exist.  
Verify that no truncation problems exist,

#### 2.3.2 Navigation

Verify tab and arrow navigation is correct.  
Verify that upon entry, the OK button is active

#### 2.3.3 OK and Close window

Verify that the OK button and the close window button shut dismiss the window and that the wizard does not start up.

#### 2.3.4 Supported types

Verify the following CPU types are supported:

- CPU222 release 1.10 and greater
- CPU224 release 1.10 and greater
- All CPU226
- All CPU226XM

#### 2.3.5 Unsupported types

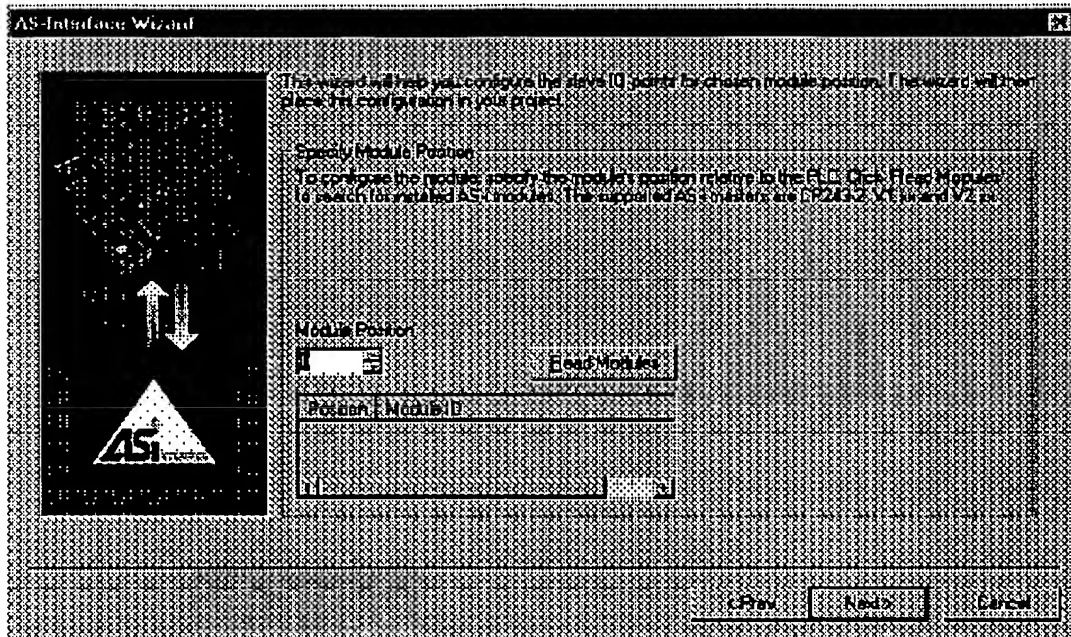
Verify the following CPU types are unsupported:

- All Gen01
- CPU221
- CPU222 release < 1.10

## AS-I Wizard Test Plan

CPU224 release < 1.10

### 2.4 Screen 2



#### 2.4.1 Appearance

Verify that the picture is correct and is located correctly within the window.  
Verify that text is technically and grammatically correct.  
Verify that no spelling errors exist.  
Verify that no truncation problems exist.

#### 2.4.2 Navigation

Verify tab and arrow navigation is correct.  
Verify that upon entry, the NEXT button is active

#### 2.4.3 Previous button

Verify that the previous button takes you back to screen 2

#### 2.4.4 Next button

Verify that the next button takes you to the next screen

#### 2.4.5 Cancel

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.  
Verify that the wizard cancels gracefully.

## AS-I Wizard Test Plan

### 2.4.6 Read modules

Attempt with no communications to the PLC

Attempt with module ready but not set

Attempt with no command acknowledgement

Attempt with error conditions returned, both known and unknown error responses.

Attempt with no modules present

Attempt with modules in each valid position

Attempt with different versions of masters connected, including less than V1.xx, V1.xx, V2.xx, and greater than V2.xx

Attempt with valid and invalid ID strings in the master

When Read module is pressed, the following window is displayed



Verify navigation is correct

Verify YES is active

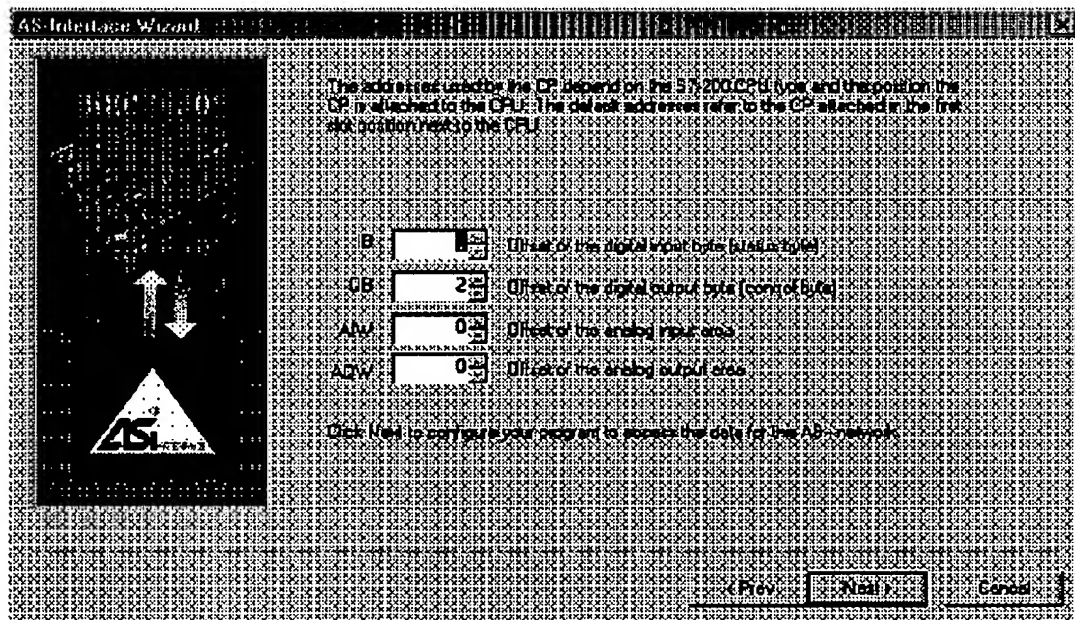
Verify NO cancels without placing the PLC into STOP

### 2.4.7 Module position spinner

Verify that the module position spinner ranges from 0 to max supported by the various controllers.

Max is 5 for most controllers (check on max for 222, think < 1.20 is 0-1)

## 2.5 Screen 3



### 2.5.1 Appearance

Verify that the picture is correct and is located correctly within the window.

Verify that text is technically and grammatically correct.

Verify that no spelling errors exist.

**AS-I Wizard Test Plan**

---

Verify that no truncation problems exist,

**2.5.2 Navigation**

Verify tab and arrow navigation is correct.

Verify that upon entry, the NEXT button is active

**2.5.3 Previous button**

Verify that the previous button takes you back to screen 2

**2.5.4 Next button**

Verify that the next button takes you to the next screen

**2.5.5 Cancel**

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.

Verify that the wizard cancels gracefully.

**2.5.6 Disable test**

Verify that when you reach this window after doing a READ modules that all spinners contain the correct value and are grayed out.

**2.5.7 IB spinner**

Verify the IB spinner works correctly between the ranges of 0-15

**2.5.8 QB spinner**

Verify the QB spinner works correctly between the ranges of 0-15

**2.5.9 AIW spinner**

Make sure the AIW spinner works correctly between the ranges of 0-62 in increments of 2 for controllers > CPU222. Ranges for the CPU222 are 0-16

**2.5.10 AQW spinner**

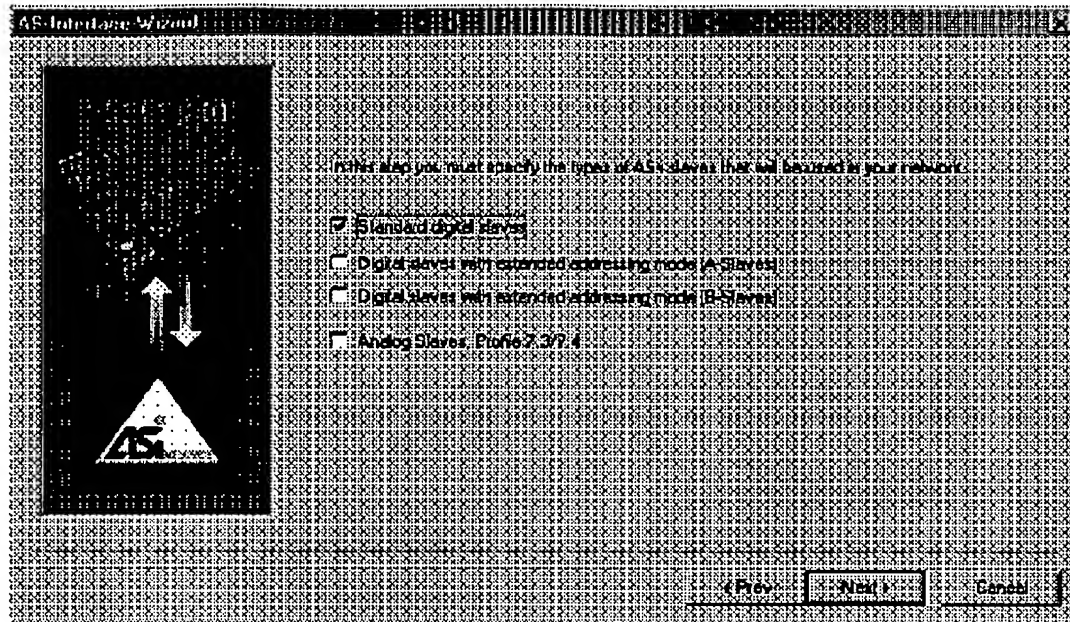
Make sure the AQW spinner works correctly between the ranges of 0-62 in increments of 2 for controllers larger than the CPU222. Ranges for the CPU222 are 0-16

**2.5.11 Read Modules fill in**

Verify that when Read modules was pressed in the previous screen that the correct ranges are placed in these fields and that they are locked.

Verify for each CPU type supported in a variety of positions.



**AS-I Wizard Test Plan****2.6 Screen 4****2.6.1 Appearance**

Verify that the picture is correct and is located correctly within the window.  
Verify that text is technically and grammatically correct.  
Verify that no spelling errors exist.  
Verify that no truncation problems exist.

**2.6.2 Navigation**

Verify tab and arrow navigation is correct.  
Verify that upon entry, the NEXT button is active

**2.6.3 Previous button**

Verify that the previous button takes you back to screen 2

**2.6.4 Next button**

Verify that the next button takes you to the next screen

**2.6.5 Cancel**

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.  
Verify that the wizard cancels gracefully.

**2.6.6 Disable test**

When an AS-I master is found with the READ modules function, it will be identified as unknown, V1.x or V2.x. Only the V2.x or above supports extended addressing and analog slaves. These will be grayed out of unsupported by the master.

**AS-I Wizard Test Plan**

---

**2.6.6.1 V BETA master**

Use the simulator to return a non 1.x or 2.x master. Verify that this case is handled correctly

**2.6.6.2 V1.x master**

Use the simulator to return a V1.x master. Verify that the Digital slaves checkbox is enabled and checked. Verify that the remaining 3 are grayed out and unchecked.

**2.6.6.3 V2.x master**

Verify that all slave boxes are enabled and checked

**2.6.6.4 No read**

Get to this window without performing a READ module function. Verify that all check boxes are enabled. Verify that the first is checked and the remaining 3 are unchecked.

**2.6.6.5 No slave selected**

Uncheck all boxes and press the NEXT button. Verify the following window is displayed. Do this again with the PREV button.

**2.6.6.6 Field locking**

Whenever a slave is entered into a grid later in the wizard, the field corresponding to that slave type becomes selected and locked.

Go forward and enter a standard slave. Back up and verify that the standard slaves field is locked.

Repeat this for A slaves, B slaves, and Analog slaves.

Try different combinations.

**2.7 Screen 5 - Digital Slave**

This screen allows the user to enter configurations for digital slaves.

There are six variants of this screen depending on whether the user has selected extended slaves.

There is much common behavior shared between each of these variants. It needs to be checked against only one variant.

**2.7.1 Shared behaviors****2.7.1.1 Appearance**

Verify that the picture is correct and is located correctly within the window.

Verify that text is technically and grammatically correct.

Verify that no spelling errors exist.

Verify that no truncation problems exist.

**2.7.1.2 Navigation**

Verify tab and arrow navigation is correct.

Verify that upon entry, the NEXT button is active

**AS-I Wizard Test Plan**

---

**2.7.1.3 Previous button**

Verify that the previous button takes you back to screen 2

**2.7.1.4 Next button**

Verify that the next button takes you to the next screen

**2.7.1.5 Cancel**

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.  
Verify that the wizard cancels gracefully.

**2.7.1.6 Invalid symbol names**

Enter invalid names within a wide range of cells, verify that they are displayed in red when you leave the cell.

Verify that you are not allowed to leave this window if invalid symbol names exist.

**2.7.1.7 Duplicate symbol names**

Enter duplicate names within a wide range of cell, verify that they are displayed in with a green squiggly line when you leave the cell.

Verify that you are not allowed to leave this window if duplicate symbol names exist.

Be sure to verify duplicates symbols are caught when duplicates exist in the following places:

Symbols within the current grid

User symbol table

Other wizard protected symbol tables

Symbol tables of AS-I modules in different positions

Symbols defined in the analog grid

Verify that all instances of the duplication are caught. For example, define a duplicate symbol in several cells and verify that they are caught.

**2.7.1.8 Creation of unique symbol names**

Whenever you drop a new slave into the grid, a unique symbol name must be created. Verify that this is so by defining symbols in all the places specified in the duplicate symbol names section.

Verify that

**AS-I Wizard Test Plan****2.7.2 Standard Slave**

Use this table to specify the slaves on your network. Specify each slave by its network address and I/O configuration. Each column in this table represents an AS slave. To add a new slave, please double click on the I/O configuration row to bring up the select dialog. Use the list to choose the slave type. Unlike default symbol names to slave IDs will be generated once the slave type is chosen. You may modify these symbol names at any time.

Address	Slave #1	Slave #2	Slave #3
I/O Configuration			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev    Next >    Cancel

Slave types supported for this variant can be found in the specification.

Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.

Verify correct behavior when an update occurs that includes Extended A and B slaves.

Verify correct behavior when an update occurs that contains analog slaves

Verify correct behavior when an update occurs that contains unknown slave types.

**2.7.2.1 Read from CPU tests****2.7.2.1.1 Valid slaves**

Verify that when slaves are read from the controller that all valid slaves are recognized in the correct positions. This includes all profiles and Ids, including extended Ids.

**2.7.2.1.2 Invalid slaves**

Verify that invalid slave types are detected and displayed for all slave locations. Invalid slave types include:

## AS-I Wizard Test Plan

### 2.7.3 Extended A slave only

Use this table to specify the slaves in your network. Specify each slave by its network address and I/O Configuration. Each column in the table represents an AS-I slave. To add a new slave, press the button on the I/O Configuration row to bring up the select dialog. Use the dialog to choose the slave type. Unlike default symbol names, slave IDs will be generated once the slave type is chosen. You may modify these symbol names in Jovian.

Address	Slave #1 A	Slave #2 A	Slave #3 A
I/O Configuration			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev   Next >   Cancel

Slave types supported for this variant can be found in the specification.

Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.

Verify correct behavior when an update occurs that includes standard and Extended B slaves.

Verify correct behavior when an update occurs that contains analog slaves

Verify correct behavior when an update occurs that contains unknown slave types.

#### 2.7.3.1 Read from CPU tests

##### 2.7.3.1.1 Valid slaves

Verify that when slaves are read from the controller that all valid slaves are recognized in the correct positions. This includes all profiles and Ids, including extended Ids.

##### 2.7.3.1.2 Invalid slaves

Verify that invalid slave types are detected and displayed for all slave locations. Invalid slave types include:

Invalid standard slaves

Invalid A slaves

## AS-I Wizard Test Plan

### 2.7.4 Extended B slave only

Address	Slave #1 B	Slave #2 B	Slave #3 B
I/O Configuration			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

Slave types supported for this variant can be found in the specification.

Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.

Verify correct behavior when an update occurs that includes standard and Extended A slaves.

Verify correct behavior when an update occurs that contains analog slaves

Verify correct behavior when an update occurs that contains unknown slave types.

#### 2.7.4.1 Read from CPU tests

##### 2.7.4.1.1 Valid slaves

Verify that when slaves are read from the controller that all valid slaves are recognized in the correct positions. This includes all profiles and Ids, including extended Ids.

##### 2.7.4.1.2 Invalid slaves

Verify that invalid slave types are detected and displayed for all slave locations. Invalid slave types include:

Any standard slave

Any Analog slave

Invalid B slaves

**AS-I Wizard Test Plan****2.7.5 Standard and extended A slave**

AS-I Wizard

Use this table to specify the slaves in your network. Specify each slave by its network address and I/O configuration. Each column in this table represents a AS-I slave. To add a new slave, please double click on the I/O configuration row to bring up the selection dialog. To change the slave type, double click on the slave name. Slaves will be generated once the slave type is chosen. You may modify these slave names in your test.

Address	Slave #1/11 A	Slave #2/12 A	Slave #3/13
I/O Configuration			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev Next > Cancel

Slave types supported for this variant can be found in the specification.

Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.

Verify correct behavior when an update occurs that includes Extended B slaves.

Verify correct behavior when an update occurs that contains analog slaves

Verify correct behavior when an update occurs that contains unknown slave types.

**2.7.5.1 Read from CPU tests****2.7.5.1.1 Valid slaves**

Verify that when slaves are read from the controller that all valid slaves are recognized in the correct positions. This includes all profiles and Ids, including extended Ids.

**2.7.5.1.2 Invalid slaves**

Verify that invalid slave types are detected and displayed for all slave locations. Invalid slave types include:

Any standard slave

Any Analog slave

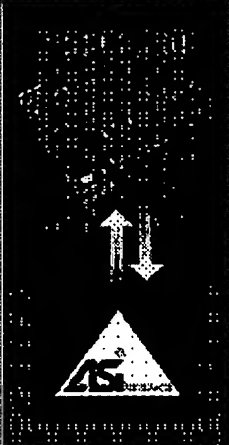
Invalid B slaves



## AS-I Wizard Test Plan

## 2.7.6 Standard and extended B slave

AS-Interface Wizard

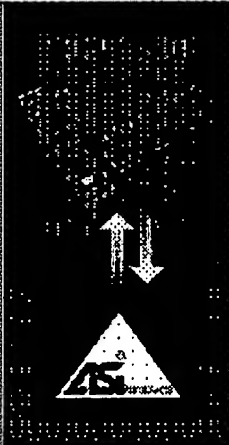


Use this table to specify the slaves on your network. Specify each slave by its network address and I/O Configuration. Each column in this table represents 1 AS-I slave. To add a new slave, please double click on the I/O Configuration row to bring up the select list. Use the list to choose the slave type. Unless default symbol names to slaves IDs will be generated once the slave type is chosen. You may modify these symbol names as you wish.

Address	Slave #1	Slave #2	Slave #3
I/O Configuration:			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev Next > Cancel

AS-Interface Wizard



Use this table to specify the slaves on your network. Specify each slave by its network address and I/O Configuration. Each column in this table represents 1 AS-I slave. To add a new slave, please double click on the I/O Configuration row to bring up the select list. Use the list to choose the slave type. Unless default symbol names to slaves IDs will be generated once the slave type is chosen. You may modify these symbol names as you wish.

Address	Slave #1	Slave #2	Slave #3
I/O Configuration:			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev Next > Cancel

Slave types supported for this variant can be found in the specification.

Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.

Verify correct behavior when an update occurs that includes Extended A slaves.

Verify correct behavior when an update occurs that contains analog slaves

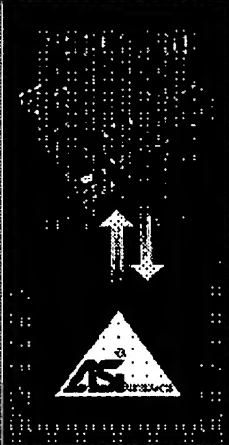
Verify correct behavior when an update occurs that contains unknown slave types.



## AS-I Wizard Test Plan

## 2.7.7 Standard with extended A and B slaves

AS-Interface Wizard

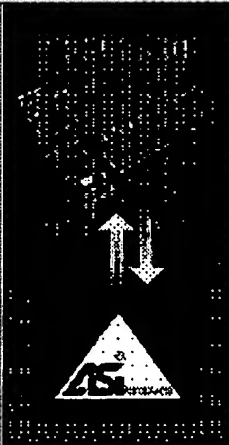


Use this table to specify the slaves on your network. Specify each slave by its network address and I/O Configuration. Each column in this table represents a AS-I slave. To add a new slave, please click on the I/O Configuration row to bring up the select list. In the list, choose the slave type. Unless default symbol names for slave I/Os will be generated once the slave type is chosen. You may modify these symbol names at your risk.

Address	Slave #1/121 A	Slave #2/122 A	Slave #3/123
I/O Configuration:			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev Next > Cancel

AS-Interface Wizard

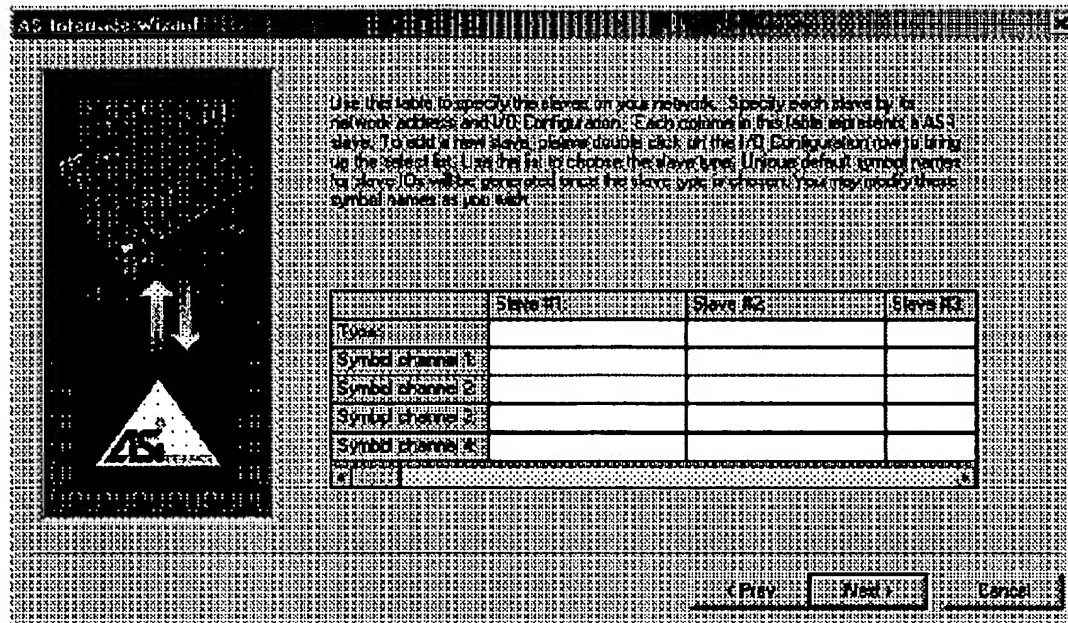


Use this table to specify the slaves on your network. Specify each slave by its network address and I/O Configuration. Each column in this table represents a AS-I slave. To add a new slave, please click on the I/O Configuration row to bring up the select list. In the list, choose the slave type. Unless default symbol names for slave I/Os will be generated once the slave type is chosen. You may modify these symbol names at your risk.

Address	Slave #1/121 A	Slave #1/121 B	Slave #2/122 B
I/O Configuration:			
Symbol Input 1			
Symbol Input 2			
Symbol Input 3			
Symbol Input 4			
Symbol Output 1			
Symbol Output 2			
Symbol Output 3			
Symbol Output 4			

< Prev Next > Cancel

Slave types supported for this variant can be found in the specification.  
 Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.  
 Verify correct behavior when an update occurs that contains analog slaves  
 Verify correct behavior when an update occurs that contains unknown slave types.

**AS-I Wizard Test Plan****2.8 Screen 6 Analog Slave****2.8.1 Appearance**

Verify that the picture is correct and is located correctly within the window.  
 Verify that text is technically and grammatically correct.  
 Verify that no spelling errors exist.  
 Verify that no truncation problems exist.

**2.8.2 Navigation**

Verify tab and arrow navigation is correct.  
 Verify that upon entry, the NEXT button is active

**2.8.3 Previous button**

Verify that the previous button takes you back to screen 2

**2.8.4 Next button**

Verify that the next button takes you to the next screen

**2.8.5 Cancel**

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.  
 Verify that the wizard cancels gracefully.

**2.8.6 Invalid symbol names**

Enter invalid names within a wide range of cells, verify that they are displayed in red when you leave the cell.

## AS-I Wizard Test Plan

Verify that you are not allowed to leave this window if invalid symbol names exist.

### 2.8.7 Duplicate symbol names

Enter duplicate names within a wide range of cell, verify that they are displayed in with a green squiggly line when you leave the cell.

Verify that you are not allowed to leave this window if duplicate symbol names exist.

Be sure to verify duplicates symbols are caught when duplicates exist in the following places:

Symbols within the current grid

User symbol table

Other wizard protected symbol tables

Symbol tables of AS-I modules in different positions

Symbols defined in the analog grid

Verify that all instances of the duplication are caught. For example, define a duplicate symbol in several cells and verify that they are caught.

### 2.8.8 Creation of unique symbol names

Whenever you drop a new slave into the grid, a unique symbol name must be created. Verify that this is so by defining symbols in all the places specified in the duplicate symbol names section.

Verify that

### 2.8.9 Update

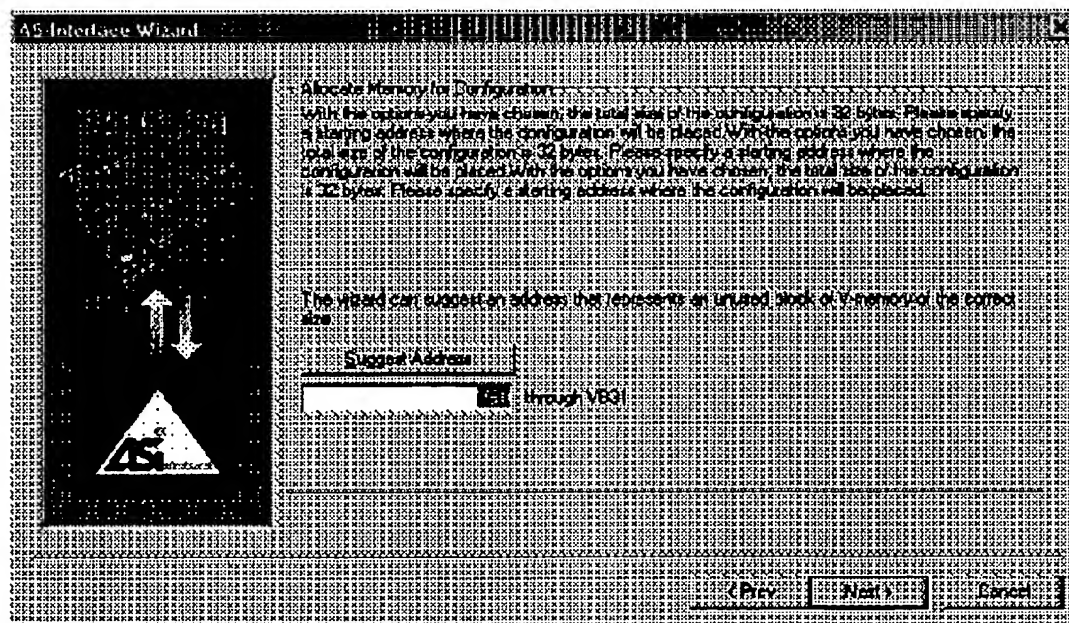
Slave types supported for this grid can be found in the specification.

Verify that all valid slaves for this type can be updated from the controller. This includes the special actuator and sensor types.

Verify correct behavior when an update occurs that includes standard, extended A and extended B slaves.

Verify correct behavior when an update occurs that contains unknown slave types.

## 2.9 Screen 7



This screen specifies where the configuration data will be place in the controller.

## **AS-I Wizard Test Plan**

---

### **2.9.1 Appearance**

Verify that the picture is correct and is located correctly within the window.

Verify that test is technically and grammatically correct.

Verify that no spelling errors exist.

Verify that no truncation problems exist.

Verify that the text accurately reports the number of VB locations required for the configuration. This should be run with the minimum sized configuration as well as the maximum and a scattering of samples between.

### **2.9.2 Navigation**

Verify tab and arrow navigation is correct.

Verify that upon entry, the NEXT button is active

### **2.9.3 Previous button**

Verify that the previous button takes you back to screen 2

### **2.9.4 Next button**

Verify that the next button takes you to the next screen

### **2.9.5 Cancel**

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.

Verify that the wizard cancels gracefully.

### **2.9.6 Address box**

Verify that only VB addresses can be specified. Try VW, QB, and a few others, both valid addresses and invalid addresses.

Verify that only addresses that are within range for the target controller can be specified. Try this with different controllers.

Verify that the correct ending address is displayed based on the starting address and the number of bytes required.

### **2.9.7 Suggest address button**

The suggest address button will run through the cross reference and look for holes large enough to fit the configuration

Create a few situations in which a number of holes, too small to fit, must be skipped over in order to find a fit.

Attempt the case in which no holes could be found.

Hit suggest a number of times until end of memory is reached. Verify that this is reported and that wrapping occurs and it begins again from the first VB locations.

## **3 Reload tests**

### **3.1 Configuration verification**

Verify that configurations created by this wizard can be reloaded without error.

Verify projects that have AS-I generated data can be saved and loaded correctly

Configurations must be saved/loaded as well as downloaded and uploaded.



**AS-I Wizard Test Plan**

---

Select delete and then cancel out of the wizard, be sure the configuration is not damaged in any way. Create at least three configurations, be sure the desired configuration is deleted without damaging the other two configurations.

**3.2.7 Move configuration**

Verify that when the Move box is unchecked that the position spinner and the read operations are blocked  
Verify that when the Move box is selected that the position spinner and the read operation are allowed.  
Move a configuration from each of the positions and verify that it is moved correctly.  
Symbol names for each configuration will be retained upon move.  
POU names will be changed.  
Make sure that everything is moved. No pieces should be left behind.  
Verify that a configuration cannot be moved into a position that already contains a configuration.

**3.2.8 Position spinner**

Verify that the position spinner is bounded by 0 at the lower end and the maximum number of expansion I/O-1 per CPU on the upper end. Try with all supported CPU types.

**3.2.9 READ module**

Verify the case where no modules are present  
Verify that the function correctly identifies Asi masters. For example, configure some modules that kind of look like Asi, like an 8DI/8DO or an 8AI/8AO by themselves.  
Find one module at each position  
Verify multiple ASI masters found. Fill the whole thing with ASI.

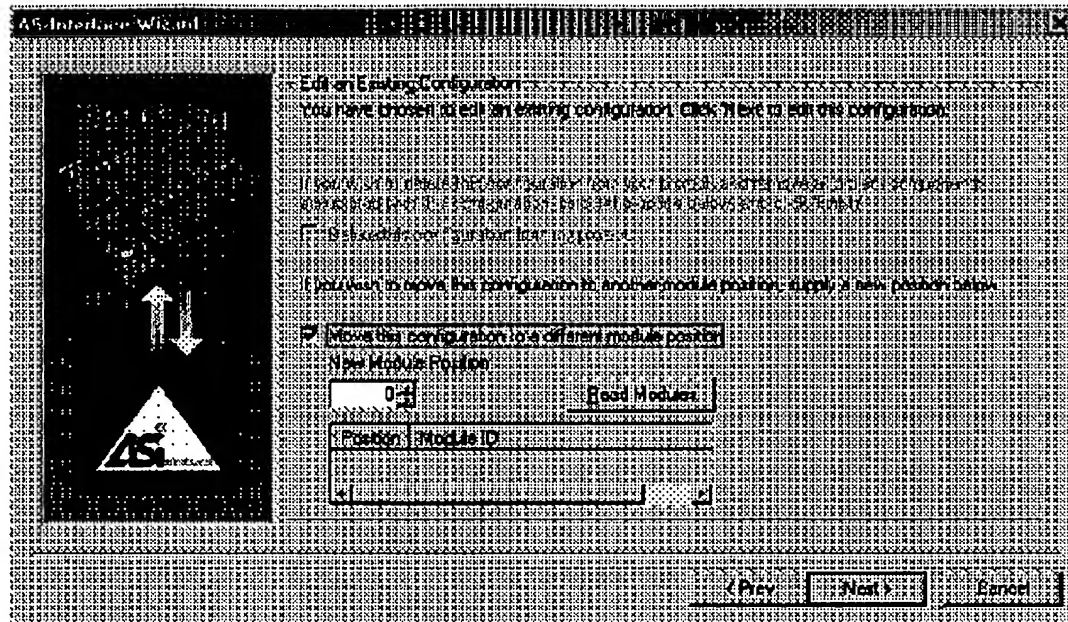
**3.2.10 Suggest address button**

The suggest address button will run through the cross reference and look for holes large enough to fit the configuration  
Create a few situations in which a number of holes, too small to fit, must be skipped over in order to find a fit.  
Attempt the case in which no holes could be found.  
Hit suggest a number of times until end of memory is reached. Verify that this is reported and that wrapping occurs and it begins again from the first VB locations.

**3.3 Deleting configuration**

Verify that delete deletes the configuration and all associated with it  
Verify that when you cancel with delete selected that nothing is deleted  
Verify that other configurations are not damaged when you delete a configuration



**AS-I Wizard Test Plan****3.4 Moving configuration**

Move a configuration from each position to each position.

Move a configuration from between two other configurations

Move a configuration into a hole between two other configurations

Attempt to move a configuration into a spot that is already occupied

Make sure that when the configuration is moved, that the location occupied previously is cleaned up correctly.

**4 Compare tests**

The compare function must be able to detect the following situations:

**4.1 Command and response bytes**

Verify with command and response byte match, also AIW/AQW

Verify with command byte different

Verify with response byte different

Verify with AIW different

Verify with AQW different

Verify with all different

**4.2 Slave match/mismatch**

Slave in Micro/WIN matches the one in the master

Slave in Micro/WIN but missing in master

Slave in master, but missing in Micro/WIN

Slave in Micro/WIN is different from that in master

**4.3 Communication disruption**

Verify case handled where master ready bit is not set

Verify case where command acknowledge never comes high

Verify error responses, both known and unknown are handled correctly

## **AS-I Wizard Test Plan**

---

Verify that compare handles communications disruptions OK.(cables pulls et. al.

### **5 Update tests**

Verify that update handles unknown slaves gracefully

Verify that update does not overwrite existing slave definitions. Warnings should be posted telling that a slave was not updated.

Verify update works correctly for all slave types in all slave positions.

### **6 Code Generation tests**

#### **6.1 Symbol tables**

Verify that symbol tables are created correctly for all slave types in all slave positions. This includes addresses as well as symbol names

#### **6.2 POU elements**

Verify that any POU elements are created correctly. Possible POU elements include:

##### **6.2.1 ASIx\_CTRL**

This subroutine is responsible for the transfer of I/O data to and from the master.

##### **6.2.2 ASIx\_READ**

This subroutine will read the specified bank from the master. This is pretty straight forward, the only thing that really needs to be verified is that the correct addresses get placed in to the code.

##### **6.2.3 ASIx\_WRITE**

This subroutine will read the specified bank from the master. This is pretty straight forward, the only thing that really needs to be verified is that the correct addresses get placed in to the code.

#### **6.3 POU comments**

Verify that all POU comments are accurate and that there are no grammatical or spelling problems.

#### **6.4 Data block**

Verify that no entries are added to the data block and that this wizard does not touch the data block in any way.

#### **6.5 IEC**

Verify that all generated POUs compile correctly when running in IEC mode.

#### **6.6 Analog Input filtering**

Verify that the correct analog inputs have filtering turned off.

### **7 Functional tests**

#### **7.1 Discrete transfer**

Discrete transfers are handled by the \_CTRL block. Whenever discrete transfers are required, be sure that the network to handle them is present in CTRL.

Standard and A slave transfers are accomplished by communicating with Bank 0 of the AS-I master.



**AS-I Wizard Test Plan**

---

**7.2 Extended address B transfer**

Discrete slave type B transfers are handled by the \_CTRL block. Whenever discrete transfers are required, be sure that the network to handle them is present in CTRL.

Slave B transfers are accomplished by communicating with Bank 31 of the AS-I master.

**7.3 Analog transfer**

Analog transfers are handled by the \_CTRL block. Whenever an analog transfer to a specific slave is configured in the grid, additional code is added to the analog section of the block. Verify that these sections are added correctly.

Analog transfers are accomplished by communicating with Banks 32-48 of the AS-I master.

**7.4 AS-I Read**

The \_READ block allows user read access to any bank within the controller.

Verify that error codes are handled correctly.

Verify that this block is functioning as expected.

**7.5 AS-I write**

The \_WRITE block allows user read access to any bank within the controller.

Verify that error codes are handled correctly.

Verify that this block is functioning as expected.

**8 Change slave address tests**

The change slave address screen allows customers to reprogram their slave addresses for online slaves.

Verify that this screen cannot be invoked unless communications to the CPU are functioning.

On entry to this function, the following screen should be displayed.

**8.1 Appearance**

Verify that the picture is correct and is located correctly within the window.

Verify that text is technically and grammatically correct.

Verify that no spelling errors exist.

Verify that no truncation problems exist.

**AS-I Wizard Test Plan**

---

Verify that the text accurately reports the number of VB locations required for the configuration. This should be run with the minimum sized configuration as well as the maximum and a scattering of samples between.

**8.2 Navigation**

Verify tab and arrow navigation is correct.  
Verify that upon entry, the NEXT button is active

**8.3 Previous button**

Verify that the previous button takes you back to screen 2

**8.4 Next button**

Verify that the next button takes you to the next screen

**8.5 Cancel**

Verify that the cancel button and the Window X button bring up the cancel failsafe dialog.  
Verify that the wizard cancels gracefully.

**8.6 Delete configuration**

Verify that when you check the delete button that the NEXT button changes to FINISH.  
Select delete and press finish. Verify both yes and no cases of the finish dialog  
Select delete and then cancel out of the wizard, be sure the configuration is not damaged in any way.  
Create at least three configurations, be sure the desired configuration is deleted without damaging the other two configurations.

**8.7 Move configuration**

Verify that when the Move box is unchecked that the position spinner and the read operations are blocked  
Verify that when the Move box is selected that the position spinner and the read operation are allowed.  
Move a configuration from each of the positions and verify that it is moved correctly.  
Symbol names for each configuration will be retained upon move.  
POU names will be changed.

**8.8 Communications tests**

Verify that broken communications is detected and handled correctly.  
Verify correct behavior when the master ready bit is off  
Verify correct operation whenever the command active bit fails to turn on  
Verify correct response to error codes, both known and unknown errors.

**8.9 Slave detection**

Verify that this window correctly detects all slave types in all positions.  
Verify for a variety of master positions  
Verify the case when all slaves are programmed  
Verify the case when no slaves are programmed  
Verify only Standard  
Verify only A  
Verify only B  
Verify only analog

**8.10 Slave moving**

Verify that standard can only be moved into standard locations(A/B check box off)  
Verify that A slaves can be moved into either A or B space

**AS-i Wizard Test Plan**

---

Verify that B slaves can be moved into either A or B space  
 Verify that analog slaves and only be moved within standard slave space  
 Verify that a slave cannot be moved on top of itself  
 Verify that a slave cannot be moved on top of an existing slave

**9 Ozzy****9.1 Introduction**

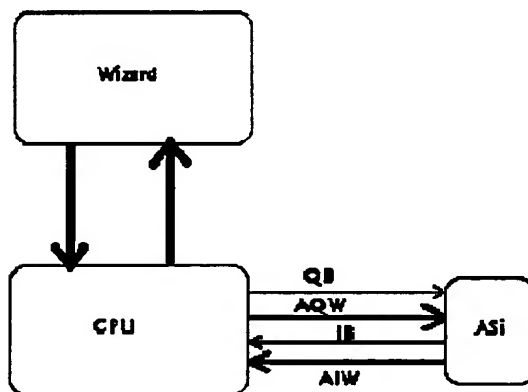
Functionality of the AS-i wizard includes the ability to read and identify slave modules from the master. For an exhaustive test of this functionality, a minimum of X different slaves need to be purchased in order to verify that the Wizard correctly identifies all slaves. In addition, many different slaves that are not supported need to be purchased.

The wizard can read up to 31 non extended and 31 additional extended slaves. This would require the purchase of 62 slaves, power supplies, and wiring to achieve comprehensive coverage. The cost to purchase and setup of all these slave is prohibitive, therefore an AS-i master simulator was created.

The AS-i master simulator (Ozzy) connects to a CPU226XM with special firmware and allows the tester to simulate every possible slave located at every possible slave address. Up to three Masters can be simulated at the same time.

**9.2 Hardware requirements:**

Traditional AS-i configuration

**Traditional ASi configuration**

In a traditional AS-i configuration, an AS-i master is connected to the S7-200 CPU as a pair of I/O modules. Micro/WIN sends commands and data to the AS-i master using eight words of AQW locations and a byte of QB memory.

The AQW locations contain the commands, typically written to bank 2, and the QB byte is used as a command byte to tell the master to process a request.

Micro/WIN then reads the command handshaking from the IB byte and the data from the AIW words.

Some commands process with a couple of iterations, some require many iterations.

The program in the S7-200 also communicates with the AS-i master using this interface.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**